

Учебный курс
**Технологии и средства разработки
корпоративных систем**

Лекция 13

**Расширенные возможности корпоративных
приложений, основы СУБД**

Лекции читает

кандидат технических наук, доцент

Зыков Сергей Викторович

Cryptography Application Block

- Облегчает решение задач, связанных с криптографией — шифрование информации, создание хэш-ключей по данным, сравнение хэш-значений для проверки целостности данных. Блок криптографии абстрагирует код приложения от криптопровайдеров: при необходимости криптопровайдеры могут быть изменены в конфигурационных файлах без изменения кода приложения и без повторной компиляции.
- Поддерживаются только симметричные алгоритмы. Асимметричные алгоритмы на основе публичных ключей, когда один ключ используется для шифрования, другой — для расшифровки, не поддерживаются.

Data Access Application Block

- Набор функций управления соединениями, создания и кэширования параметров хранимых процедур и т.п. Базируется на принципе применения абстрактных провайдеров данных (ADO .NET 2.0) на уровне классов DbCommand и DbConnection для получения параметров и преобразования типов. Т.о., приложения могут быть перенесены из одного хранилища данных в другое без модификации исходного кода.
- Классы с функциональностью, специфичной для Microsoft SQL Server, Microsoft SQL Server CE и Oracle.
- Обращение к базе данных по имени строки соединения. Код приложения может создать именованный экземпляр базы данных и передать этот параметр методу DatabaseFactory.CreateDatabase.
- Каждая именованная БД имеет информацию о соединении, хранимую в конфигурационном файле. Изменяя эту информацию, разработчики могут использовать приложения с различными конфигурациями БД без перекомпиляции приложения.

Exception Handling Application Block

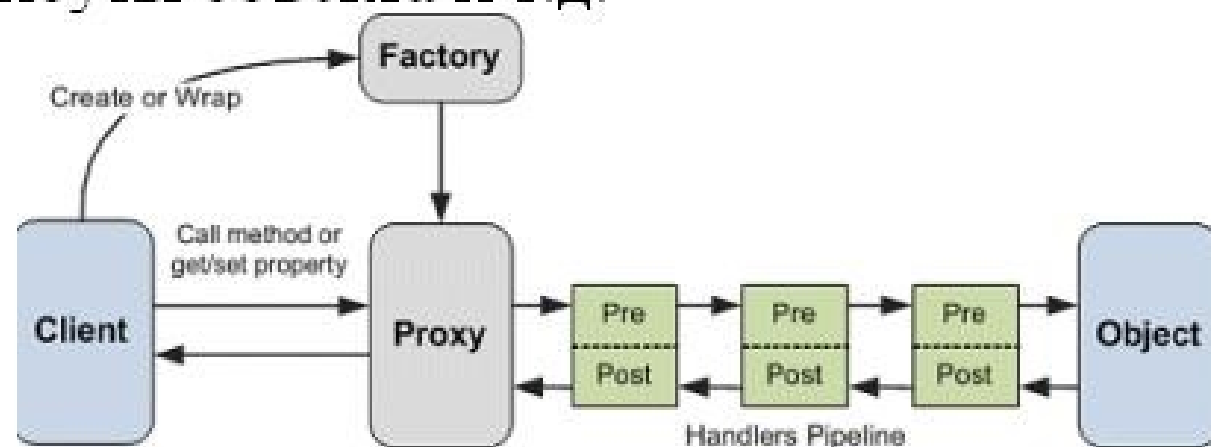
- Унифицирует обработку исключений на всех уровнях приложения. За счет внешней (по отношению к приложению) конфигурации разработчики и администраторы могут выбрать способ обработки исключений. Предоставляет механизмы для протоколирования исключений, замены одного исключения другим, сохранения контекстной информации посредством помещения одного исключения внутрь другого, для создания собственных способов обработки исключений.
- Определяет *политики обработки исключений*, обеспечивающие связь между определенным классом исключений и действиями по их обработке.

Logging Application Block

- Реализует механизмы ведения протоколов.
- Разработчики могут использовать блок протоколирования для записи информации в Event Log (штатный протокол событий на уровне операционной системы) для передачи данных по электронной почте, сохранения данных в базе данных, в очереди сообщений и в текстовом файле, для генерации WMI-события или в других местах, задаваемых через точки расширения данного функционального блока.

Policy Injection Application Block

- Разработчики могут устанавливать правила, регламентирующие выполнение различных операций (регистрация данных, кэширование, обработка ошибок и подтверждение достоверности информации) внутри приложения.
- Для выбора конкретного объекта приложения правила можно указать широкий диапазон особенностей, в т.ч. имя сборки, пространства имен, тип, имя объекта, атрибуты объекта и т.д.



Security Application Block

- Реализует механизмы авторизации и безопасного кэширования данных, связанных с авторизацией и аутентификацией.
- Функциональность целиком базируется на библиотеке классов .NET Framework.

Unity Application Block

- «Легкий», расширяемый контейнер
- Функции:
 - добавление зависимостей,
 - добавление конструкторов,
 - добавление полей и методов.

Validation Application Block

- Интегрирован с Windows Forms, ASP.NET и Windows Communication Foundation
- Позволяет встраивать в приложения механизмы проверки достоверности данных.

Пример использования Data Access (1)

Чтобы использовать Enterprise Library, необходимо добавить сборки `Microsoft.Practices.EnterpriseLibrary.Data` и `Microsoft.Practices.EnterpriseLibrary.Configuration` в проект (right-click на References, Add Reference, \Program Files\Microsoft Enterprise Library\bin; добавить файлы `Microsoft.Practices.EnterpriseLibrary.Data.dll` и `Microsoft.Practices.EnterpriseLibrary.Configuration.dll`)

- Затем, добавить в начало кода класса две директивы Imports:
- `Imports Microsoft.Practices.EnterpriseLibrary.Data`
`Imports Microsoft.Practices.EnterpriseLibrary.Data.Sql`
- Далее надо добавить в обработчик событий Page_Load следующий код:

Пример использования Data Access (2)

- Imports
Microsoft.Practices.EnterpriseLibrary.Data.Sql
- **Добавить в обработчик событий Page_Load следующий код (Visual Basic):**

```
Dim db As Database = DatabaseFactory.CreateDatabase()  
Dim sqlCommand As String = ... <SQL-запрос, подобный:>  
    "SELECT * FROM TableName" ...  
Dim dbCommandWrapper As DBCommandWrapper =  
    db.GetSqlStringCommandWrapper(sqlCommand)  
DataGridID.DataSource =  
    db.ExecuteReader(dbCommandWrapper)  
DataGridID.DataBind()
```

Microsoft SQL Server 2008

Содержание

1. Введение – История и технологии Microsoft SQL Server
2. Основные возможности
 - Службы аналитики
 - Анализ данных
 - Схема сетевой готовности
 - Службы интеграции
 - Управляемость
 - Производительность и масштабируемость
 - Программируемость
 - Службы отчетов
 - Безопасность
 - Пространственные данные
3. Библиография

История Microsoft SQL Server

- 1992 — SQL Server 4.2
- 1993 — SQL Server 4.21 под Windows NT
- 1995 — SQL Server 6.0, кодовое название SQL95
- 1996 — SQL Server 6.5, кодовое название Hydra
- 1999 — SQL Server 7.0, кодовое название Sphinx
- 1999 — SQL Server 7.0 OLAP, кодовое название Plato
- 2000 — SQL Server 2000 32-bit, кодовое название Shiloh (версия 8.0)
- 2003 — SQL Server 2000 64-bit, кодовое название Liberty
- 2005 — SQL Server 2005, кодовое название Yukon (версия 9.0)
- 2008 — SQL Server 2008, кодовое название Katmai (версия 10.0)

Корпоративные системы
Корпоративные приложения и СУБД

SQL Server как центр данных



Технологии Microsoft SQL Server

1. Службы аналитики
2. Анализ данных
3. Схема сетевой готовности
4. Службы интеграции
5. Управляемость
6. Производительность и масштабируемость
7. Программируемость
8. Службы отчетов
9. Безопасность
10. Пространственные данные